VIEW-DEPENDENT NON-UNIFORM SAMPLING FOR IMAGE-BASED RENDERING

Cha Zhang and Tsuhan Chen

Dept. of Electrical and Computer Engineering, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
{*czhang, tsuhan*}@*andrew.cmu.edu*

ABSTRACT

In this paper, we propose an algorithm for view-dependent non-uniform sampling for image-based rendering (IBR). Given a set of virtual views, the positions of the capturing cameras are rearranged in order to obtain the optimal rendering quality. The resulting arrangement of the cameras is effectively non-uniform sampling of the plenoptic function. We formulate the above sampling problem as a recursive weighted vector quantization problem, which can be solved efficiently. Experimental results show that the non-uniform sampling scheme renders much better images than traditional uniform sampling methods.

1. INTRODUCTION

Image-based rendering (IBR) has received much attention recently [1]. From the signal processing point of view, IBR is the sampling and interpolation of the 7D plenoptic function [2], which describes all the light rays at any position (3D), along any direction (2D), at any time (1D) and over any wavelength (1D). Compared with the old way of 3D rendering which relies heavily on accurate geometric models, IBR is easy to capture and fast to render. Most importantly, IBR can generate very realistic views unparalleled by the traditional model-based rendering.

One disadvantage of IBR, however, is the huge amount of images required for alias-free rendering. In the literature, representative IBR techniques such as light field rendering [3] and concentric mosaics [4] often use over-sampling to avoid aliasing artifacts. Chai et al. [5] presented the first formal uniform sampling analysis on the light field for scenes with Lambertian surface and without occlusions. Their work was later extended by Zhang and Chen [6] to scenes with non-Lambertian surface and occlusions, as well as concentric mosaics. Unfortunately, according to the theoretical sampling analysis, the minimum number of images needed by a typical scene is in the order of thousands, which brings practical concerns during the capturing and storage of the data.

To reduce the overall number of images needed by IBR, Zhang and Chen [7] proposed the Position-Interval Error (PIE) function for the non-uniform sampling of IBR. The PIE function measures the error of interpolation given the location and interval of the samples in a certain neighborhood. As a general framework for non-uniform sampling, PIE leads to two practical algorithms specifically designed for IBR: progressive capturing (PCAP) and rearranged capturing (RCAP). PCAP captures the scene by progressively adding cameras at the places where PIE are maximal. RCAP, on the other hand, assumes that the overall number of cameras is fixed and tries to rearrange the cameras such that the PIE has equal value everywhere. With non-uniform sampling, it is possible to achieve good rendering results from hundreds of images, which is a big improvement over uniform sampling.

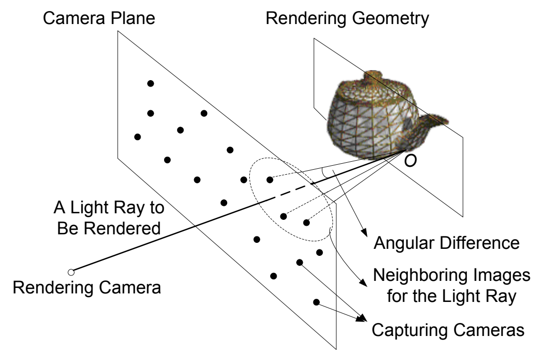

Fig. 1. The capturing and rendering of IBR scenes.

In this paper, we propose a view-dependent non-uniform sampling scheme which aims at further reducing the number of images required by IBR. We make an additional assumption, i.e., during the capturing, the virtual view's rendering position is known. This assumption is practical if the rendering process is performed on-the-fly during the capturing process. We show that the cameras can be rearranged such that the quality of the virtual view is optimized. The camera positions are computed using a recursive weighted vector quantization (VQ) method, which can be solved efficiently.

The paper is organized as follows. Section 2 briefly reviews the typical capturing and rendering process of IBR scenes. Section 3 presents the proposed algorithm for view-dependent non-uniform sampling. Experimental results are described in Section 4. Conclusions are presented in Section 5.

2. IBR SCENE CAPTURING AND RENDERING

Fig. 1 shows a typical IBR system including capturing and rendering. We place a set of cameras around the object and shoot images. In the light field [3] setup, the cameras are uniformly distributed on a plane (namely the camera plane), and point to the same direction. In the concentric mosaics [4], the cameras are arranged along a circle. In the most general form, the cameras can be anywhere, which can still be rendered through the unstructured Lumigraph rendering [8]. In this paper, we assume that the cameras are constrained on a camera plane, but on that plane the distribution of the cameras can be non-uniform. The directions of the cameras are assumed to be the same, although this requirement is not crucial to the proposed algorithm.

To render novel views from the captured images, we split the virtual view into many light rays and obtain their intensities one by one. As shown in Fig. 1, consider one of the light rays being rendered. We first trace the light ray back to the scene geometry, and obtain the crossing point *O*. As the geometry is typically unknown for real-world scenes, usually a constant depth plane is assumed. We then project *O* to the neighboring captured images (circled by an ellipse in Fig. 1) and obtain the light ray's intensity through weighted interpolation of all the projections. The weights are usually determined by the angular difference between the rendered light ray and the projection directions from *O* to the captured images, which also serves as an criterion for selecting the neighboring images. The smaller the angular difference, the higher the weight to the associated image. Other factors such as resolution or field of view may also affect the weights [8], however they are not considered in the current implementation.

3. VIEW-DEPENDENT IBR NON-UNIFORM SAMPLING

3.1. Problem Statement

The view-dependent non-uniform sampling problem can be stated as follows. Assume we have *N* cameras to capture a static or slowly-moving scene. They can move freely on the same camera plane, and point to the same direction. During the capturing, we also have *P* viewers who are watching the scene. These *P* views are rendered through the above mentioned method from the *N* captured images. The goal is to arrange these *N* cameras such that the *P* views can be rendered at their best quality. Here both *N* and *P* are finite.

The above problem becomes trivial if all the *P* views are on the camera plane, and $P \leq N$, because one may simply move the cameras to the virtual view positions and capture the scene at those places directly. However, the problem is valid as long as one of the virtual viewpoints are out of the camera plane (even if $P = 1$), or $P > N$. This is because any out of plane view will have to be synthesized from multiple images, which can be potentially improved by rearranging the capturing cameras.

3.2. Formulation Based on the Angular Difference

Let the cameras' positions on the camera plane be $\mathbf{c}_j, j = 1, 2, \cdots, N$. For the *P* views being rendered, we may split them into totally *L* light rays. Denote them as $l_i, i = 1, 2, \cdots, L$. The intersection of the light rays and the camera plane are denoted as $\mathbf{x}_i, i = 1, 2, \cdots, L$. As shown in Fig. 2, consider a certain light ray l_i, which crosses the scene geometry at *O*, and one of its neighboring camera $\mathbf{c}_j$. Denote the distance between $\mathbf{c}_j$ and $\mathbf{x}_i$ as d_{ij} and the angular difference as θ_{ij}. Let the distance between *O* and $\mathbf{x}_i$ be r_i, which is known beforehand. From the figure, we know that when the scene depth $r_i \gg d_{ij}$ (which is often true), we have:

$$\theta_{ij} \approx \frac{d_{ij} \cos \alpha_i}{r_i} = w_i \|\mathbf{x}_i - \mathbf{c}_j\| \quad (1)$$

where α_i is the angle between the light ray l_i and the normal of the camera plane (not shown in Fig. 2), and $w_i = \frac{\cos \alpha_i}{r_i}$. Let

$$\widetilde{\theta}_i = \min_{j=1,\cdots,N} \theta_{ij} \quad (2)$$

be the minimum angular difference between light ray l_i and its neighboring images. A sufficient (but not necessary[1]) condition

[1]This condition is not necessary because even if $\widetilde{\theta}_i$ is large, the rendering quality can still be good if the scene geometry is accurate and the scene surface is Lambertian.

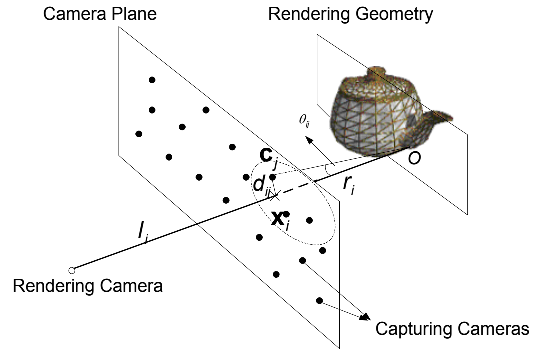

Fig. 2. The formulation of the problem.

that the light ray l_i will be rendered correctly, is that $\widetilde{\theta}_i$ is very small. Therefore, to have a very good rendering quality of all the light rays, we may minimize $\sum_{i=1}^{L} \widetilde{\theta}_i$. Given Equ. 1 and 2, the best camera positions can be obtained as:

$$\widehat{\mathbf{c}_j} = \arg\min_{\mathbf{c}_j} \sum_{i=1}^{L} w_i \min_{j=1,\cdots,N} \|\mathbf{x}_i - \mathbf{c}_j\| \quad (3)$$

for $j = 1, \cdots, N$. As $\mathbf{x}_i$ and w_i is known during the rendering (the scene depth r_i and light ray direction α_i are known), Equ. 3 is a standard weighted VQ problem, and can be easily solved.

3.3. Formulation Based on the Local Color Consistency

One constraint about the algorithm proposed in the last subsection is that angular difference is only a sufficient but not necessary condition of good rendering quality. This means that if we do use $w_i = \frac{\cos \alpha_i}{r_i}$ as the weight, we may not get optimal rendering quality by minimizing Equ. 3. However, if the w_i in Equ. 3 can be defined based on a better criterion, we may still use the same weighted VQ algorithm to achieve the optimal rendering quality.

We propose to use the *local color consistency* as a better condition for good rendering quality. The local color consistency was first proposed in [7] for measuring the PIE functions. It states that a good rendering quality can be expected if the projections used to interpolate a certain light ray share the same intensity. Therefore, an easy implementation of the local color consistency is to use the variance of the projections. Take the light ray l_i in Fig. 2 as an example. The local color consistency $\mathcal{C}_i$ of l_i can be calculated as:

$$\mathcal{C}_i = \frac{1}{\sigma_i} \quad (4)$$

where σ_i is the variance of the projections of *O* to the *k* nearest neighboring images. Usually *k* is small, e.g., $k = 4$. In [7] it has been shown that such measurement is indeed a good estimate of the rendering quality.

It is therefore natural to think of w_i as a function of $\mathcal{C}_i$. That is:

$$w_i = f(\mathcal{C}_i) \quad (5)$$

Here $f(\cdot)$ is a function which produces a large w_i when $\mathcal{C}_i$ is small, vise versa.

Unfortunately, unlike the minimization of Equ. 3 using $w_i = \frac{\cos \alpha_i}{r_i}$ in the last subsection that has a clear physical meaning (minimize the minimal angular difference for all the light rays to their closest camera), defining w_i as in Equ. 5 can be rather arbitrary and does not have a physical meaning. In the next subsection, we propose an algorithm that finds w_i recursively aiming at making all the $\mathcal{C}_i$ to be large as well as equal to each other.

3.4. A Recursive Algorithm for View-Dependent Non-Uniform Sampling

In the previous sections, we have been vaguely using the word "best rendering quality" for the goal of non-uniform sampling. Here we give a more well-defined goal:

> *A non-uniform sampling scheme is optimal if the local color consistency of all the light rays share the same value, which is as large as possible.*

The above goal is similar to the one we used in [7]. It is targeted to make the rendered light rays have equal quality everywhere. Such property is desired so that the rendered views deliver a constant quality over the whole images. Meanwhile, we try to make the consistency value be as large as possible, which improves the overall quality of the rendering.

Having set the goal of non-uniform sampling, let us examine how to achieve such a goal. In Equ. 3, if the weight w_i is given, the minimization process will try to find the best locations of the capturing cameras such that their weighted distance to the rendered light rays is minimized. As the local color consistency usually increases when such distance reduces, the rendering quality will be improved as a whole. However, such minimization process does not affect whether the resultant local color consistency values are equal or not. To make the values equal, one must adjust the weight w_i. Next, we propose an recursive two-stage algorithm that can achieve the goal.

Fig. 3 shows the flow chart of our proposed non-uniform sampling algorithm applicable for static or slowly-moving scenes. Given a set of newly captured images, we may first measure the local color consistency of all the light rays $l_i, i = 1, \cdots, L$ as $\mathcal{C}_i, i = 1, \cdots, L$. If we are informed that the viewers have moved to some new viewpoints during the capturing of the images, we will give some arbitrary initial values for the weights $w_i, i = 1, \cdots, L$. For instance, we can let:

$$w_i = \frac{1}{L} \quad (6)$$

Otherwise, we will simply use the weights in the last time instance. We then examine the values of the local color consistency for all the light rays. If they are equal, nothing needs to be performed, and we are ready to capture the next set of images. Otherwise, we perform an update of the weights based on the local color consistency values or the variance of the projections for the light rays. In the currently implementation, we first define:

$$s_i = \log \sigma_i \quad (7)$$

as the score for each light ray. Let s_{min} and s_{max} be the minimum and maximum value of $s_i, i = 1, \cdots, L$. $\overline{s}$ be the average value of s_i. The weight w_i^{k+1} at time instance $k+1$ is updated from those w_i^k at time instance k as:

$$w_i^{k+1} = \begin{cases} w_i^k * (1 + (\xi - 1)\frac{\overline{s} - s_i}{\overline{s} - s_{min}}), & s_i \leq \overline{s}; \\ w_i^k * (1 + (\zeta - 1)\frac{s_i - \overline{s}}{s_{max} - \overline{s}}), & s_i > \overline{s}. \end{cases} \quad (8)$$

where ξ and ζ are the minimum and maximum weight scaling factor. They are set as 0.5 and 4 respectively in the current implementation. Equ. 8 basically says that if the variance of the projections to the neighboring images for a light ray is greater than the average (thus the local color consistency is bad), its weight will be

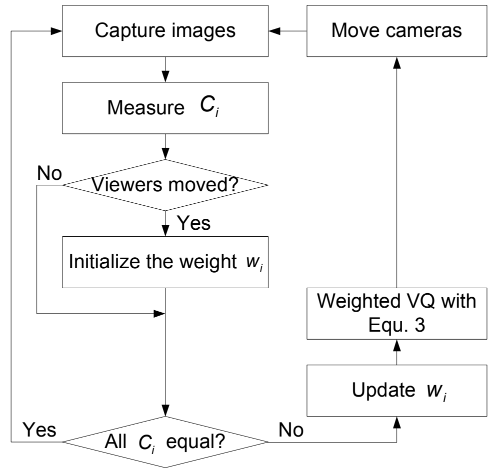

Fig. 3. The flow chart of our proposed non-uniform sampling algorithm for static or slowly-moving scenes.

increased. During the weight VQ, the camera positions will then move closer to that light ray. Otherwise, the camera positions will move away. Notice that after the weight update with Equ. 8, one should normalize the new weights such that $\sum_{i=1}^{L} w_i^{k+1} = 1$.

Having determined the new weights, we then perform a weighted VQ as Equ. 3 to obtain the new camera positions. To find the solution of Equ. 3, an slightly modified LBG-VQ algorithm [9] is adopted, which is based on the following two criteria:

1. Nearest neighbor condition:

$$\mathbf{x}_i \in \mathcal{R}_j, \text{if } \|\mathbf{x}_i - \mathbf{c}_j\| \leq \|\mathbf{x}_i - \mathbf{c}_{j'}\|, \forall j' = 1, \cdots, N \quad (9)$$

where $\mathcal{R}_j$ is the neighborhood region of centroid $\mathbf{c}_j$.

2. Centroid condition:

$$\mathbf{c}_j = \frac{\sum_{\mathbf{x}_i \in \mathcal{R}_j} w_i \mathbf{x}_i}{\sum_{\mathbf{x}_i \in \mathcal{R}_j} w_i}, j = 1, \cdots, N \quad (10)$$

The modified LBG-VQ algorithm iteratively applies Equ. 9 and Equ. 10 to find the solution of Equ. 3. The initial camera positions of the modified LBG-VQ algorithm are the current camera positions. After solving Equ. 3, the capturing cameras are then moved to the new positions and take some new images. Since the scene is static or slowly-moving, we assume that by the time the cameras move to the new positions, the scene has not changed too much.

4. EXPERIMENTAL RESULTS

We verify the effectiveness of the proposed view-dependent non-uniform sampling algorithm with a static synthetic scene, as shown in Fig. 4. The scene is named *Teapot* and is captured by 64 cameras. A single virtual view is used whose position is off the camera plane. In Fig. 4(i-a), we use a constant depth plane at the teapot's mouth as the geometric model. The body and lid of the teapot are thus blurred due to the inaccurate geometry. Fig. 4(i-b) shows the projections of the cameras' positions to the rendered view (red dots). The white triangles are used during the rendering for texture mapping, however the corner of the triangles have the same depth since we do not have the scene geometry. Fig. 4(i-c) and (i-d) are the non-uniform sampling results after one iteration of weighted

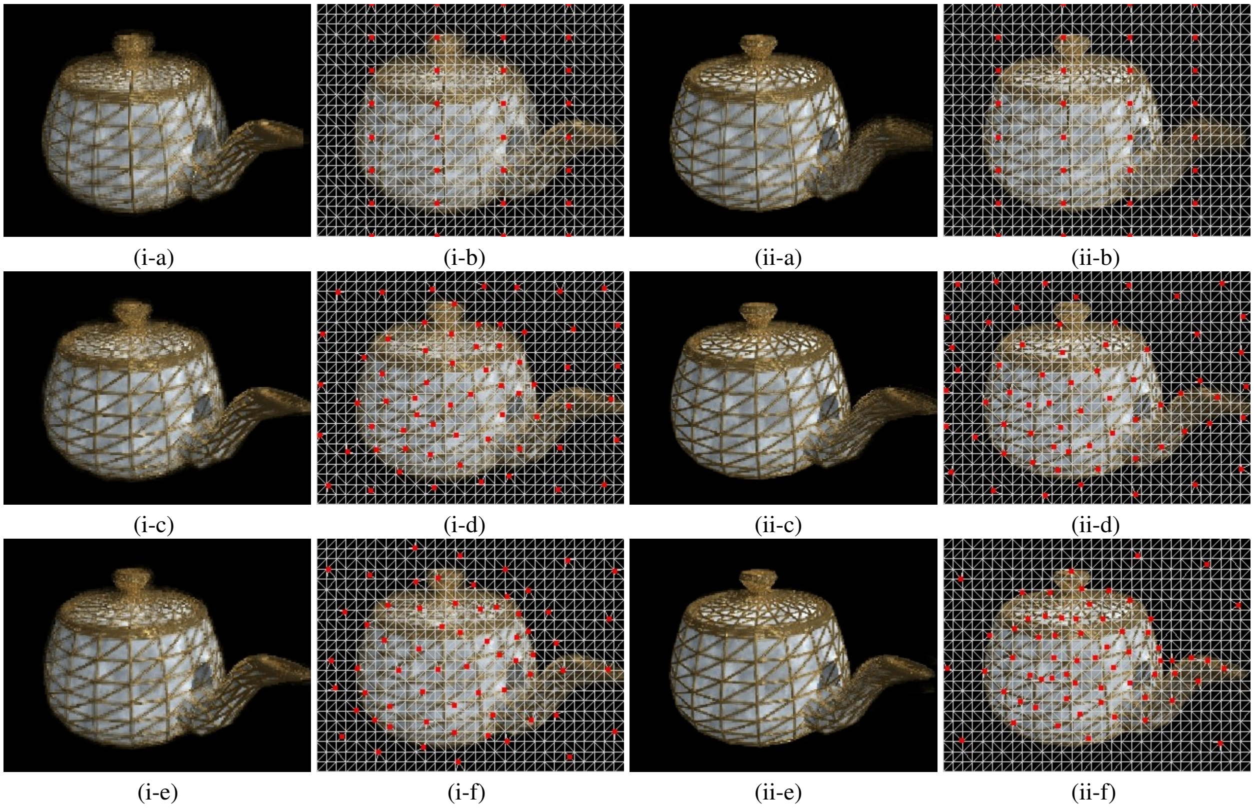

Fig. 4. Results of our view-dependent non-uniform sampling algorithm on the *Teapot* scene. (i) View rendered with the depth plane at its mouth. (ii) View rendered with the depth plane around the lid. (a)(b) Uniform sampling and its camera distribution. (c)(d) After one iteration of weighted VQ and the corresponding camera distribution. (e)(f) After 3 iterations.

VQ. The rendering quality improvement is very obvious. Notice that the camera positions are moving towards the body and lid of the teapot, where the rendering quality was bad. Fig. 4(i-c) and (i-d) are the results after 3 iterations. More samples are around the lid because that is the place that has the wrongest geometry. Fig. 4(ii-a) to (ii-f) is another set of results when the rendering depth is around the lid. Notice that the non-uniform sampling scheme automatically moves the cameras to the region of body and mouth.

5. CONCLUSIONS

This paper presented a view-dependent non-uniform sampling algorithm for IBR, which is able to improve the rendering quality of the given virtual views significantly. We believe non-uniform sampling is very promising in reducing the number of images required by IBR.

6. ACKNOWLEDGEMENT

This work is supported in part by NSF Career Award 9984858.

7. REFERENCES

[1] C. Zhang and T. Chen, "A Survey on Image-Based Rendering - Representation, Sampling and Compression," *EURASIP Signal Processing: Image Communication*, pp. 1-28, Vol. 19, No. 1, Jan. 2004.

[2] E. H. Adelson, and J. R. Bergen, "The Plenoptic Function and the Elements of Early Vision," *Computational Models of Visual Processing*, Chapter 1, Edited by Michael Landy and J. Anthony Movshon. The MIT Press, Cambridge, Mass. 1991.

[3] M. Levoy and P. Hanrahan, "Light Field Rendering," *Computer Graphics (SIGGRAPH'96)*, pp. 31-42, Aug. 1996.

[4] H.Y. Shum and L.-W. He, "Rendering with Concentric Mosaics," *Computer Graphics (SIGGRAPH'99)*, pp.299-306, Aug. 1999.

[5] J.X. Chai, X. Tong, S.C. Chan and H. Y. Shum, "Plenoptic sampling," *Computer Graphics (SIGGRAPH'00)*, pp.307-318, July 2000.

[6] C. Zhang and T. Chen "Spectral Analysis for Sampling Image-Based Rendering Data," *IEEE Trans. on CSVT*, pp. 1038-1050, Vol. 13, No. 11, Nov. 2003.

[7] C. Zhang and T. Chen, "Non-Uniform Sampling of Image-Based Rendering Data with the Position-Interval Error (PIE) Function," *Visual Communication and Image Processing (VCIP) 2003*, Lugano, Switzerland, Jul. 2003.

[8] C. Buehler, M. Bosse, L. McMillan, S. Gortler and M. Cohen, "Unstructured Lumigraph rendering," *Computer Graphics (SIGGRAPH'01)*, pp 425-432, Aug. 2001.

[9] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. on Communications*, pp. 702-710, Jan. 1980.